2022 Fall Student Project: Deep Supervised Learning for MRI Reconstruction

Weijie Gan and Yuyang Hu

July 14, 2022

Background: Consider Magnetic Resonance Imaging (MRI) reconstruction as an inverse problem that recovers an image $\boldsymbol{x} \in \mathbb{C}^n$ from a noisy measurement $\boldsymbol{y} \in \mathbb{C}^n$ characterized by a linear model

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{F}\boldsymbol{x} + \boldsymbol{e} \;, \tag{1}$$

where $\boldsymbol{F} \in \mathbb{C}^{n\times n}$ denotes the Fourier transform, $\boldsymbol{P} \in \mathbb{C}^{n\times n}$ represents a sampling operator, and $\boldsymbol{e} \in \mathbb{C}^{n\times n}$ is a noise vector. In many cases, we only use the magnitude of $\boldsymbol{x}$ for diagnosis.

Deep learning (DL) has gained popularity in MRI reconstruction due to its excellent performance. A common strategy relies on supervised learning that trains a convolutional neural network (CNN) $f_{\boldsymbol{\theta}}$ by mapping zero-filled images $\{\widehat{\boldsymbol{x}}_i = \boldsymbol{F}^{-1}\boldsymbol{y}_i\}_i^N$ to their desired ground-truth $\{\boldsymbol{x}_i\}_i^N$. Here, $\boldsymbol{F}^{-1}$ denotes inverse Fourier transform, and N is the total number of training samples. The loss function is formulated as

$$\frac{1}{N}\sum_i^N \|f_{\boldsymbol{\theta}}(\widehat{\boldsymbol{x}}_i) - \boldsymbol{x}_i\|_2^2 \;. \tag{2}$$

This student project is to implement the supervised learning method for MRI reconstruction.

Step-by-Step Instructions:

- Download dataset.h5 from https://drive.google.com/file/d/1qp-l9kJbRfQU1W5wCjOQZi7I3T6jwA37/view?usp=sharing. Use trnOrg as the set of ground-truth $\{\boldsymbol{x}_i\}$ and trnMask as the corrsponding $\{\boldsymbol{P}_i\}$.
- Obtain $\{\boldsymbol{y}_i\}$ and $\{\widehat{\boldsymbol{x}}_i\}$ as discussed above. Set $\{e_i\}$ to zero. Visualize $\{\widehat{\boldsymbol{x}}_i\}$.
- Divide $\{\widehat{\boldsymbol{x}}_i\}$ and $\{\boldsymbol{x}_i\}$ into training, validation and testing sets. Implement the supervised training as formulated in (2). Evaluate the performance by computing PSNR and SSIM values of the testing data. The PSNR and SSIM value can be compuated with scikit-image package.
- Please use Colab if GPU resource is needed.

Presentation requirement:

- Briefly introduce the data pre-processing procedure and the neural network structure you used.
- Make a table with average quantitative results in the testing set.
- Make a figure that illustrates a groundtruth selected from the testing set, the corresponding zero-filled image and the corresponding reconstructed image side by side. Put the corresponding PSNR and SSIM values above the images.
- (Optional) We realize that creating a Github repo and putting everything (codes, tables, and figures) into a Jupyter notebook should be a good choice for debugging and presentation.

Good to think about: We summarize the key points of this project here, which the student should think about them carefully before starting the implementation

- Data normalization (having the data in the same scalar range) is in general important in machine learning. How are you going to normalize your data?
- Unlike many vision tasks, data in MRI reconstruction are complex values, but many existing functions in many Python libraries does not support complex values natively (*e.g.,* convolutional layer in Pytorch). How are you going to handle it?